Engineering Applications of Artificial Intelligence xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

One-time password authentication scheme based on the negative database

Dongdong Zhao[a,b], Wenjian Luo[a,*]

[a] Anhui Province Key Laboratory of Software Engineering in Computing and Communication, School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China
[b] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, Hubei, China

ARTICLE INFO

Keywords:
One-time password
Authentication
Negative database
One-way hash function

ABSTRACT

In this paper, a novel one-time password authentication scheme based on the negative database (*NDB*) is proposed. The authentication data, which involve a user password and random number, are converted to an *NDB* before they are transmitted to the network. Recovering the original database (*DB*) from an *NDB* is an *NP*-hard problem. Even if the data transmitted in the network have been intercepted by an attacker, the attacker cannot recover the password due to the hardness of reversing the *NDB*. The proposed scheme is the first one-time password authentication scheme based on the *NDB*. Following the method used in this paper, the *NDB* can be added to other authentication schemes as an extra layer to further improve security. The proposed scheme can be adopted into other applications such as business management, network-based consumer electronics, and intelligent household systems.

1. Introduction

In recent years, along with the rapid development of network technologies, system security, legitimate network resource use, and privacy preservation have been widely studied. Authentication technology is an important security control mechanism. Most traditional authentication technologies are based on static passwords. Static password authentication is one of the simplest and most efficient authentication technologies; however, it is not highly secure because the password constantly remains the same. Static password authentication cannot resist replay attacks, password leakage, guessing attacks, or exhaustive attacks. To overcome the defects of static password authentication, a one-time password authentication is proposed (Lamport, 1981). In this scheme, the password or authentication data sent to the server is different in each login request. It can resist replay attacks, guessing attacks, and exhaustive attacks. Password leakage can be prevented because the password is usually encrypted or transformed by one-way functions.

Thus far, many one-time password authentication schemes have been proposed to protect the authentication data and user password. The security of the schemes proposed in early years was primarily based on information known by the user (Joyce, 1990; Kim, 1999), such as the one-time password authentication scheme based on a one-way hash function proposed by Lamport (1981) and its improved versions (Haller, 1995; Sandirigama et al., 2000; Chen and Ku, 2002). One-time password authentication schemes using smart cards were subsequently proposed in later studies (Yang and Shieh, 1999; Liao et al., 2006; Chang and Wu, 1991; Xu et al., 2009; Wu et al., 2012). These schemes depend not only on the information known by the user, but also on the smart cards that are independent from computers. Owing to these two factors, the security levels of these schemes are greatly improved, however, the implementation and computational cost are also significantly increased.

One-time password authentication schemes are usually evaluated based on three aspects, i.e., simplicity, security and efficiency. Simplicity denotes how simple it is to realize the one-time password authentication scheme. Moreover, it is related to the implementation cost, and whether the scheme is easy to use. Simplicity could additionally influence the practicality of the scheme. Security involves how difficult it is for an attacker to break the scheme and obtain the private information. Efficiency describes the computational or communication cost of the scheme. Obviously, these three criteria evaluate the effectiveness of the scheme from three different aspects.

Most existing one-time password authentication schemes are based on one-way hash functions, encryption algorithms, or several hard problems, such as the discrete logarithm problem. The schemes based on one-way hash functions, such as S/KEY (Haller, 1995) and SAS (Sandirigama et al., 2000), are usually simple and efficient (in terms of the computational cost); however, their security is not high. The schemes based on encryption algorithms, such as (Jan and Chen, 1998; Kim et al., 2008; Hwang and Li, 2000), usually have relatively high security; however, they are not efficient because they have a

* Corresponding author.
E-mail address: wjluo@ustc.edu.cn (W. Luo).

http://dx.doi.org/10.1016/j.engappai.2016.11.009
Received 29 February 2016; Received in revised form 20 November 2016; Accepted 23 November 2016
Available online xxxx
0952-1976/

relatively high computational cost. The security of encryption algorithm schemes is usually based on the discrete logarithm problem; however, additional computations are added, e.g., the computations for key management. The schemes based on hard problems, such as the discrete logarithm problem (McCurley, 1990), are usually as highly secure as the encryption algorithm schemes; however, they are more efficient in terms of computational cost (Xu et al., 2009; Lee and Chiu, 2005).

In this paper, we introduce a new security technique called negative databases (*NDBs*) to one-time password authentication, and we propose a novel one-time password authentication scheme. The *NDB* (Esponda et al., 2004a, 2004b, 2005) is a different system for data security from above traditional techniques (e.g., hash functions and encryption algorithms), and its searching space could be more complex. The proposed scheme provides two-layer authentication data protection to enhance the security without increasing much computational cost. The scheme proposed in this paper is based on the information known by the user (i.e., the password). Moreover, it can be extended to a scheme with smart cards. That is to say, the smart card can be embedded as a device that stores the private data of the user and executes all the procedures on the client side. Moreover, the proposed scheme can be adopted in applications such as business management, network-based consumer electronics, and intelligent household systems. The *NDB* has several promising properties. Such properties could be used to extend the proposed scheme, and an example is shown in Section 4. Furthermore, following the method used in the proposed scheme, the *NDB* can be added to other authentication schemes as an extra layer to further improve security. Overall, our contributions are:

(1) We propose the first one-time password authentication scheme based on the *NDB*. We analyze the security of the proposed scheme and show that it can resist replay attacks, password leakage, guessing attacks, and exhaustive attacks. Moreover, we show that our scheme is robust to message blocking, and can be extended to be secure against a man-in-the-middle attack.
(2) We analyze the computational complexity of the proposed scheme and show that it is more efficient than those schemes based on the encryption algorithms or discrete logarithm problem. We discuss the application of the proposed scheme to business management.

The rest of this paper is organized as follows: Section 2 introduces related work regarding the *NDB*, its generation algorithm, and authentication systems based on the *NDB*. Section 3 presents the proposed one-time password authentication scheme based on the *NDB*. Security and efficiency in terms of space and computational time complexity are also analyzed in Section 3. An application of the proposed scheme to business management, the hash function selection, and possible extensions of the proposed scheme are discussed in Section 4. In Section 5, we conclude this paper and discuss future work.

2. Related work

2.1. Negative database

The proposed authentication scheme is based on the *NDB*. For convenience, the *NDB* is introduced in this subsection. For a database (*DB*) with m entries $DB = \{x_1, \ldots, x_m\}$, each entry in *DB* is a binary string with length l: $x_i \in \{0, 1\}^l$. The universal set is $T = \{0, 1\}^l$, and the *NDB* stores the complementary set of *DB*. Because the number of strings in the complementary set of *DB* is usually very large, the *NDB* cannot be expressed exactly. Therefore, a "do not care" notation, denoted by "*", is introduced to compress the *NDB*. Given a string defined upon the alphabet {0, 1, *}, the positions with value "0" or "1" are defined as specified positions, and positions with "*" are defined as unspecified positions. The symbol "*" represents either "0" or "1" at given positions. An example of the *NDB* is given as follows.

$DB = \{000,\ 001\}$,
$T - DB = \{010,\ 011,\ 100,\ 101,\ 110,\ 111\}$,
A possible $NDB = \{1**,\ *1*\}$ *or* $\{1**,\ 01*\}$.

As shown in the example, the size of the compressed *NDB* could be much smaller than the size of the exact complementary set of the *DB*. Because there are some redundant expressions in the compressed *NDB*, different *NDB*s can be generated from the same *DB*. This is an important property of the *NDB*. If an *NDB* covers all the entries in the complementary set of the *DB*, the *NDB* is said to be complete, otherwise, it is incomplete. If an *NDB* can be reversed to obtain the *DB* in polynomial time, the *NDB* is said to be easy-to-reverse, otherwise, it is hard-to-reverse.

Thus far, several *NDB* applications have been researched, and examples of the studies are listed in Table 1. Reversing the *NDB* to obtain the corresponding *DB* is an *NP*-hard problem, therefore, it can be used to protect data privacy (Esponda et al., 2004b, 2009, 2007). The *NDB* can be employed to hide information by mixing the hidden data with a large number of superfluous entries (Esponda, 2008). The negative survey (Esponda, 2006) is a significant research branch of the *NDB*. In the negative survey, an interviewee is asked to select an option different from the true one with a certain probability (Esponda, 2006). According to statistical methods, the desired population proportions can be estimated from the negative survey. Horey et al. (2007) employed the negative survey to realize anonymous data collection in sensor networks. Recently, the negative survey was used by Groat et al. (2011, 2012) to preserve data privacy in a wireless sensor network and participatory sensing applications. Liu et al. (2013) proposed a privacy preserving k nearest neighbor classification algorithm and k-means clustering algorithm based on the *NDB*. Additionally, the *NDB* was applied to secure multi-party computation in (Zhao and Luo, 2013a) and privacy preserving data publication in (Du et al., 2014; Zhao and Luo, 2013b). Further, it was applied to authentication systems in (Bringer and Chabanne, 2010; Zhao et al., 2016a; Dasgupta and Azeem, 2007, 2008; Dasgupta et al., 2014). These particular studies are discussed in detail in subsection 2.3.

Table 1
Related work regarding the negative database.

Author	Application
Esponda et al. (2004b, 2009, 2007)	Protecting data privacy
Esponda (2008)	Hiding information
Esponda (2006)	Negative survey
Horey et al. (2007)	Anonymous data collection in sensor networks
Groat et al. (2011)	Privacy preserving data aggregation in wireless sensor networks
Groat et al. (2012)	Protecting data privacy in participatory sensing applications
Bringer and Chabanne (2010) Zhao et al. (2016a)	Secure biometric recognition
Liu et al. (2013)	Privacy preserving data mining
Du et al. (2014), Zhao and Luo (2013b)	Privacy preserving data publication
Zhao and Luo (2013a)	Secure multiparty computation
Dasgupta and Azeem (2007, 2008), Dasgupta et al. (2014)	Authentication system

2.2. Generation algorithm of the negative database

Several algorithms have been proposed for generating *NDB*s in recent years, such as the prefix algorithm (Esponda et al., 2004b, 2009), *RNDB* algorithm (Esponda et al., 2004b, 2009), q-hidden algorithm (Esponda et al., 2007; Jia et al., 2005), hybrid-*NDB* algorithm (Liu et al., 2011), p-hidden algorithm (Liu et al., 2014), M-hidden algorithm (Liu et al., 2015), and K-hidden algorithm (Zhao